Keldin Maldonado

Professor Paul Conrad

Assembly Programming

12 June 2020

The Ladder Game

I decided for my final project in the assembly programming course to do a game called Ladder. In the Ladder game, the player must reach the top of a ladder that is represented by five LED lights on a breadboard.

The breadboard is connected to a Raspberry Pi 3 running an OS based on the Debian branch of GNU/Linux. The Ladder game also includes a single button that lets the user provide input to the code running the Ladder game. Furthermore, the game is as follows; an LED for the current step in the ladder turns on, and the user has a short period of time to click the single button to give feedback to the code running the Ladder game that the button was pressed. However, the game only accepts user input/feedback when the light is turned on, not before or after. The LED is turned on for a couple of seconds at a time and the user must click the button. Failure to click the button in the time, and the game completely resets. If the game resets the player gets knocked down to the bottom of the ladder and must restart from the bottom. However, if the user/player can reach the top when he/she reaches the top the game does a sort of celebration for the player on the last LED. The last LED in the Ladder game is not like any other LED. The last LED is an RGB LED that can cycle through red green blue and it can also do a combination of those colors. In this instance, the last LED while playing the game represents a green color. Like previously mentioned, the game rewards the user with a celebration once the

player reaches the last LED. This celebration is demonstrated using the RGB LED mentioned previously, the LED is set to cycle through red, green, and blue.

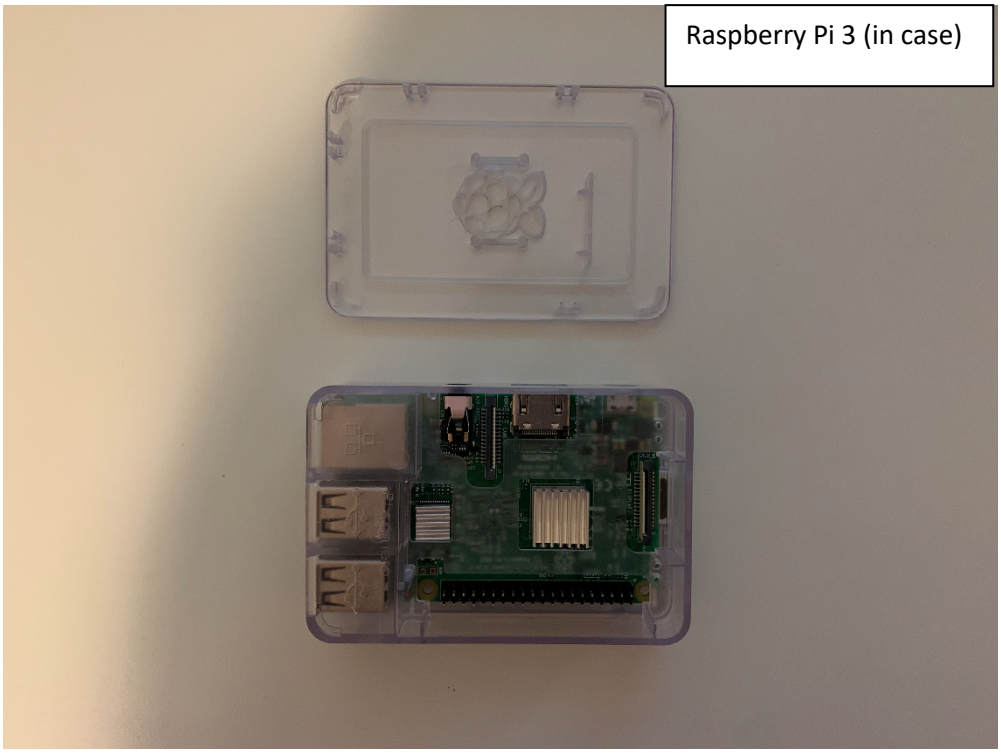
Within the Ladder game's code there are a variety of different constants that make the code behind the Ladder game run efficiently and effectively. At the top of the code, INPUT, OUTPUT could be found. These constants are used to easily set whether a pin is going to be set to the output of zero (INPUT) or one (OUTPUT). There is also a LOW, HIGH constants that hold the value that will set a pin to on or off. Furthermore, the BUTTON constant strictly holds the pin value where buttons input will be sent to according to the wiringPi library. The SECONDS constant simply holds the amount of delay the light is set to be to HIGH (on) for. Lastly, the RED, BLUE, YELLOW, BLUE_TWO, RGB_RED, RGB_GREEN, and RGB_BLUE simply hold the pin number corresponding to those LED colors according to the wiringPi layout.

To continue, the next part of the code that is important to be discussed is the different functions that make the Ladder game work. The functions in the code are main, lightsShow, checkButton, buttonRead, pinOn, pinOff, pinInput, and pinOutput. The lightsShow function is simply a scrolling and flashing function that was added to give the Ladder game a fun and fresh look to it. checkButton on the other hand uses the SECONDS constant as a sort of timer to keep the current LED to HIGH (on) for the specified SECONDS constant. Very closely linked to the checkButton function is the buttonRead function. If the buttonRead function check the button activity from the user and it returns a one (1) then the game proceeds otherwise the game resets. The functions pinOn and pinOff simply include code that either set a particular pin to HIGH (on) or LOW (off) respectively, pinInput and pinOutput do the exact same thing but with whether a pin should be accepting input or sending output.

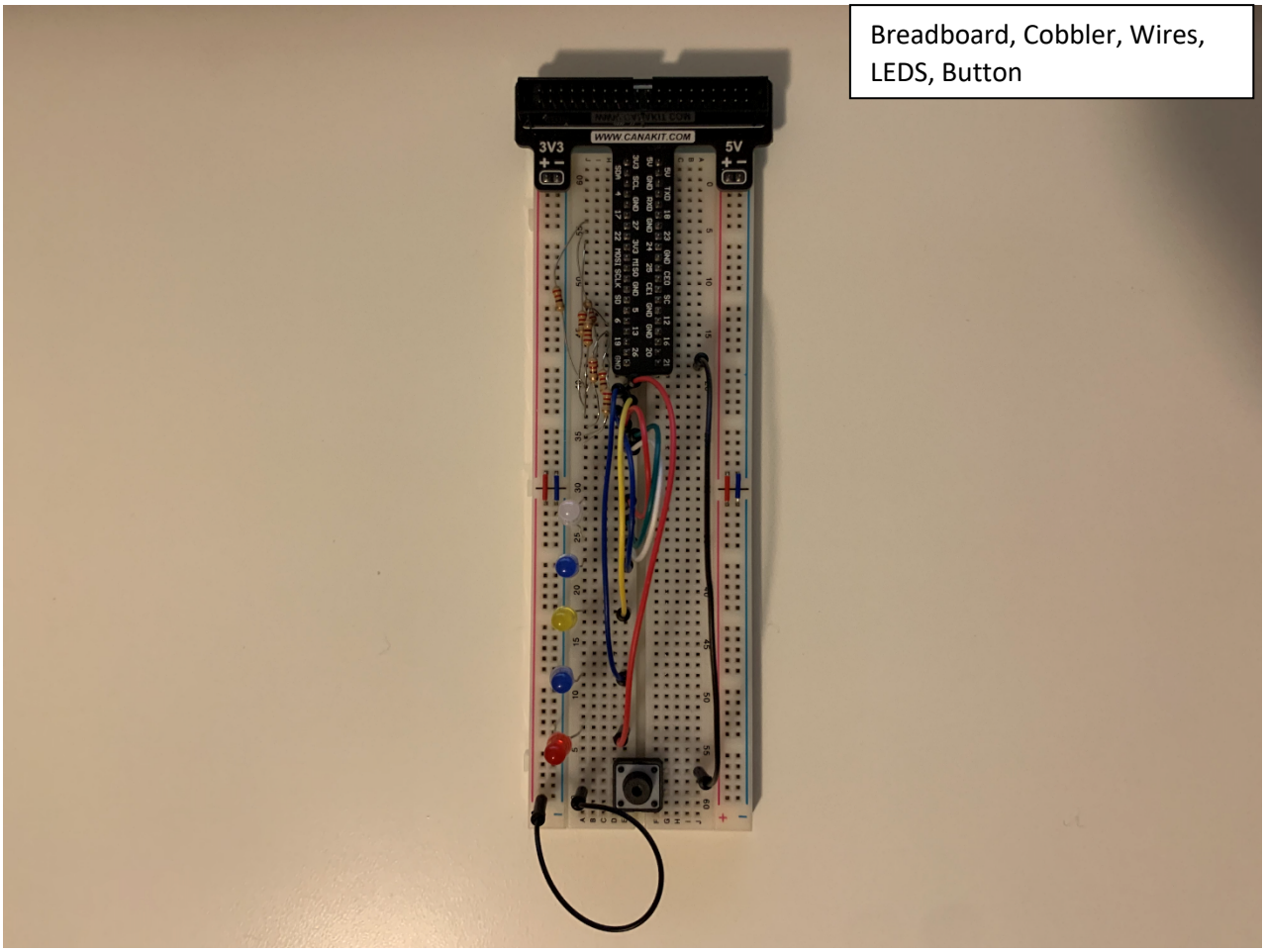
Some of the struggles that were faced when writing the code for this game was getting my mind around how to essentially build a timer for when the code reached the checkButton function. However, after some work and research I was able to get my mind around the concept of comparing the amount of seconds since January 1970 twice to be able to determine the amount of second that had elapse and then compare it to the SECONDS constant.

This is only the first version of this Ladder game, I do, however, have some ideas on how to improve it with more time available for this project. One aspect that I could not figure out is how to make the LED come on at random times instead of the set two seconds that it is currently set to. I think this could be achieved with some sort of random number generator. I think introducing the notion of the LEDs coming on at a random point say between one and ten seconds can add more of a challenge to the player/user and make the game more fun and engaging. The last aspect of this Ladder game I was not completely satisfied with is the check the code does to be able to make sure a user is not just continuously hitting the button even before an LED is on to be able to easily get to the top. A check for this was implemented in this current version of the Ladder game, however, it can be improved to be more sensitive and pick up on this aspect of cheating in future versions of the game.

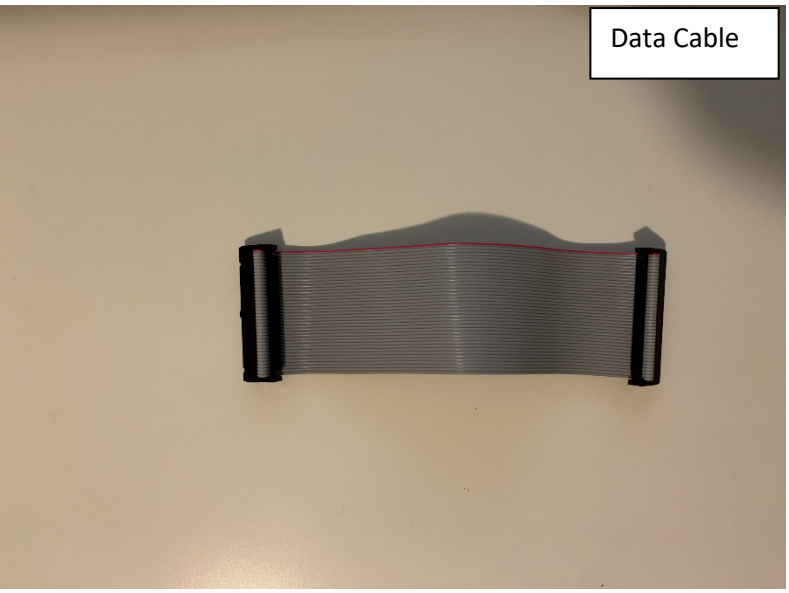
Overall, writing this program/game was extremely fun and it thought me a lot of how processors work and the vary low levels. I think being able to get down in the mud with the code and wrestling with it at nights thought me some valuable lessons in terms of troubleshooting, solving problems, and implementing code. I also think this project helped me think in a much more logical manner than I previously was thinking.

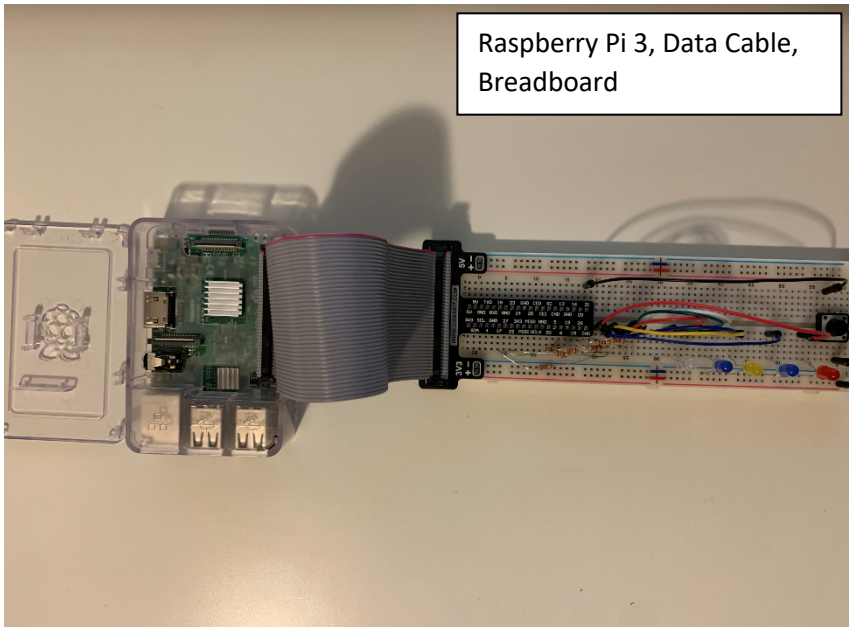


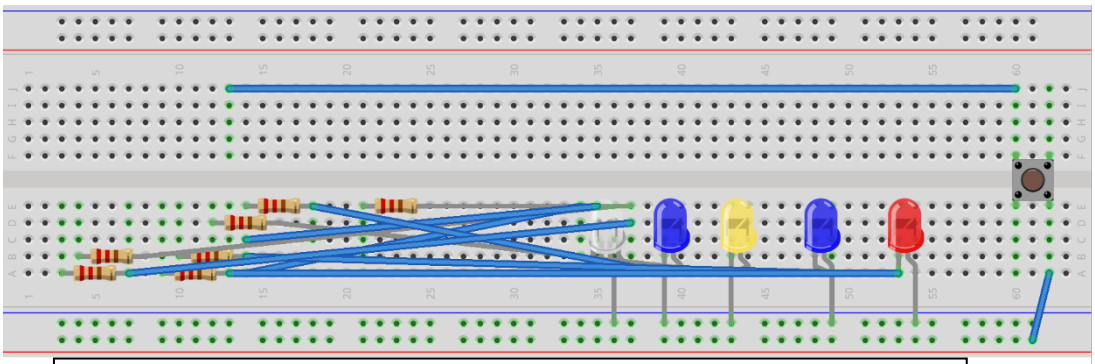
Raspberry Pi 3 (in case)



Breadboard, Cobbler, Wires, LEDS, Button


Data Cable


Raspberry Pi 3, Data Cable, Breadboard



Note: Diagram not 100% accurate could not find cobbler in fritzing